PATENT
Attorney Docket No.: 40101 - 07301 (2000.034)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	)	
	)	
Mark A. Stevens et al.	)	
	)	
Serial No.: 09/766,335	)	Group Art Unit: 2178
	)	Confirmation No. 7723
Filed: January 19, 2001	)	Examiner: Cong-Lac T. Huynh
	)	
For: CONVERSION SYSTEM FOR TRANSLATING STRUCTURED DOCUMENTS INTO MULTIPLE TARGET FORMATS	)	**Board of Patent Appeals and Interferences**
	)	

Mail Stop: Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REPLY BRIEF UNDER 37 C.F.R. § 41.41

In response to the Examiner’s Answer mailed on November 28, 2008 to the Appeal Brief filed October 20, 2008, and pursuant to 37 C.F.R. § 41.41, Appellants present this Reply Brief in the above-captioned application.

This is an appeal to the Board of Patent Appeals and Interferences from the Examiner's final rejection of claims 1, 4-20, and 22-38 in the Final Office Action dated May 22, 2008. The appealed claims are set forth in the attached Claims Appendix.

1. Status of the Claims

Claims 1, 4-20, and 22-38 have been rejected in the Final Office Action. Claims 2, 3, and 21 were canceled in a previous amendment. The final rejection of claims 1, 4-20, and 22-38 is the subject of the present appeal.

2. Ground of Rejection to be Reviewed on Appeal

I. Whether Claims 1, 4-20 and 22-38 are anticipated under 35 U.S.C. § 102(e) by U.S. Pub. App. No. 2002/0052893 to Grobler et al. (hereinafter "Grobler").

3. Argument

I. The Rejection of Claims 1, 4-20 and 22-38 As Anticipated By Grobler Should Be Reversed.

A. The Examiner's Rejection

In the Final Office Action, the Examiner rejected claims 1, 4-20 and 22-38 under 35 U.S.C. § 102(e) as anticipated by Grobler. (See 5/22/08 Office Action, pp. 2-8.) The Examiner affirmed this rejection in the Advisory Action. (See 8/11/08 Advisory Action, pp. 1-2.)

B. Grobler Does Not Disclose Identifying a Feature Set Of a Source File, Assembling the Feature Set In a Buffer, and Writing the Feature Set Into a Target File In the Target Format As Recited In Claim 20.

In responding to the Appeal Brief, the Examiner disputes the contention that "the identifying of the feature set in the source file is done prior to being stored in a buffer." (See 11/28/08 Answer, p. 9.) In disputing this contention, the Examiner argues that "the claim does not require where to perform the identifying step and moving the feature set into a buffer after being identified. The identifying step can be performed anywhere and only the assembling of the feature is required to be performed in the buffer." (Id., p. 9.) Appellants disagree. Claim 20 recites "identifying a feature set of a source file." The claim then recites "assembling the feature set in a buffer." The "source file" designates a first memory location, and the assembling step designates a buffer that should not be read to encompass the source file. Thus, the first two steps of claim 20 recites two memory locations that are distinct from each other: the "source file" and the "buffer." This is supported by Figure 1 and its accompanying description, which shows source file 110 and buffer 116 as different memory locations. One of ordinary skill in the art, when reading the claim in light of the specification, would thus disagree with the Examiner that "the claim does not require where to perform the identifying step."

Moreover, it is also true that the assembling step occurs after the identifying step. Although the ordinary rule of construction for method claims does not require that the steps of the claim be performed in the order in which they are written, in this case the language requires such a reading. The assembling step makes an antecedent reference to "the feature set." One of ordinary skill in the art would read this assembling step as, in effect, "assembling the feature set that has already been identified in the identifying step." This interpretation finds further support from the use of the term "feature set." The use of "set" suggests that the features are organized into a collection. The features are organized into a collection only by first being identified, since from the standpoint of simple logic, the features must be identified first before being regarded as belonging to a set. If that is the case, then it follows that the step of identifying the feature set precedes the step of assembling the feature step in the buffer. This is in contrast to Grobler, which moves the entire source file (called "source data") into temporary storage and analyzes it for tags, instead of following the claimed method of identifying the feature set (from a "source file" separate from the "buffer") and only then assembling the feature set in a buffer.

The Examiner even agrees that the description of Grobler in the preceding sentence is accurate. (See 11/28/08 Answer, p. 9.) ("The examiner agrees about this disclosure of the invention.") This aspect of the Grobler system is unlike that of the claimed invention, because a source file is not a buffer, and if the source file is not the buffer, then the identifying of the feature set occurs outside of the buffer, which is not taught by Grobler. Accordingly, reversal of the rejection is respectfully requested.

4. Conclusion

For the reasons set forth above, Appellants respectfully request that the Board reverse the final rejections of the claims by the Examiner under 35 U.S.C. § 102(e) and indicate that claims 1, 4-20 and 22-38 are allowable.

Respectfully submitted,

Date: January 22, 2009

By: ____________________

Michael J. Marcin (Reg. No. 48,198)

Fay Kaplun & Marcin, LLP
150 Broadway, Suite 702
New York, NY 10038
Tel: (212) 619-6000
Fax: (212) 619-0276

CLAIMS APPENDIX

1. (Previously presented) A translator, embodied in a computer readable storage medium, for translating a source file in a source format to a target file in a target format, the translator comprising:

a feature identifier to determine a feature set of the source file;

a buffer to assemble the feature set;

a feature writer to write the feature set into the target file in the target format; and an output module to output the target file.

2. (Canceled)

3. (Canceled)

4. (Previously presented) The translator of claim 1, wherein features of the feature set are selected from the group consisting of paragraph style, straddled cells in a table, cross-referencing, pen styles in a drawing, other document formatting, document header specifications, document footer specifications, discontinuity indicators, order indicators, location indicators, beginning indicators, ending indicators, data types, data translation pairs, document macros, user-created features, implied feature endings and combinations thereof.

5. (Original) The translator of claim 1, wherein the feature identifier comprises a front-end converter to map code fragments of the source file to a list of features.

6. (Original) The translator of claim 5, wherein the feature identifier comprises a front-end lookup table.

7. (Original) The translator of claim 6, wherein the front-end lookup table is user modifiable.

8. (Original) The translator of claim 1, wherein the feature writer comprises a back-end converter to map the feature set to code fragments of the target file format.

9. (Original) The translator of claim 8, wherein the back-end converter comprises a back-end lookup table.

10. (Original) The translator of claim 5, comprising a plurality of feature writers to write the feature set into a plurality of target files having a plurality of target formats.

11. (Original) The translator of claim 1, comprising a plurality of feature identifiers to determine a feature set of a plurality of source files having a plurality of source formats.

12. (Original) The translator of claim 5, wherein the front-end converter comprises a lexical analyzer to identify tokens disposed within the source file, and a feature collector to associate the tokens with the feature set.

13. (Original) The translator of claim 1, further comprising a user interface.

14. (Original) The translator of claim 13, wherein the user interface comprises a GUI.

15. (Original) The translator of claim 1, further comprising a source format adapter module to interface with a source file generator.

16. (Original) The translator of claim 15, wherein the source format adapter module enables the source file generator to initiate translation by the translator.

17. (Original) The translator of claim 1, further comprising a target file adapter module to perform secondary translation.

18. (Original) The translator of claim 17, wherein the target file adapter module translates the target file into another target format.

19. (Original) The translator of claim 1, wherein the source and target formats are selected from the group consisting of MIF, RTF, WordPerfect, VENTURA, Microsoft Word, Interleaf, HTML, SGML, SML, C, C++, Visual Basic, Pascal, Java, MFC, PowerPlant, Swing, SVG, HPJ, Flash, WMF, VRML, RenderMan, 3DMF, and combinations thereof.

20. (Previously presented) A method of translating a file from a source format to a target format, the method comprising:

(a) identifying a feature set of a source file;

(b) assembling the feature set in a buffer; and

(c) writing the feature set into a target file in the target format.

21. (Canceled)

22. (Previously presented) The method of claim 20, wherein features of the feature set are selected from the group consisting of paragraph style, straddled cells in a table, cross-referencing, pen styles in a drawing, other document formatting, document header specifications, document footer specifications, discontinuity indicators, order indicators, location indicators, beginning indicators, ending indicators, data types, data translation pairs, document macros, user-created features, implied feature endings and combinations thereof.

23. (Original) The method of claim 20, wherein the identifying step (a) comprises mapping code fragments of the source file to a feature list.

24. (Original) The method of claim 23, wherein the identifying step (a) comprises looking up the code fragments in a front-end lookup table.

25. (Original) The method of claim 24, further comprising permitting the front-end lookup table to be user modifiable.

26. (Original) The method of claim 20, wherein the writing step (b) comprises mapping the feature set to code fragments of the target file format.

27. (Original) The method of claim 26, wherein the writing step (b) comprises looking up the feature set in a back-end lookup table.

28. (Original) The method of claim 20, wherein the writing step (b) comprises writing the feature set into a plurality of target files having a plurality of target formats.

29. (Original) The method of claim 20, wherein the identifying step (a) comprises identifying a feature set of a plurality of source files having a plurality of source formats.

30. (Original) The method of claim 20, wherein the identifying step (a) comprises identifying tokens disposed within the source file, and associating the tokens with the feature list.

31. (Original) The method of claim 20, further comprising using a source file generator to initiate translation by the translator.

32. (Original) The method of claim 20, further comprising using a target file adapter module to perform secondary translation.

33. (Original) The method of claim 32, wherein the target file adapter module translates the target file into another target format.

34. (Previously presented) A method of configuring a system to translate a source file in a source format to a target file in a target format, the method comprising:

(a) providing a feature identifier to determine a feature set of the source file;

(b) providing a buffer to assemble the feature set; and

(c) providing a feature writer to write the feature set into the target file in the target format.

35. (Previously presented) A system for translating a source file in a source format to a target file in a target format, the system comprising:

a feature identifier to determine a feature set of the source file;

a buffer to assemble the feature set;

a feature writer to write the feature set into the target file in the target format; and

an output module to output the target file.

36. (Previously presented) An article of manufacture for translating a source file in a source format to a target file in a target format, the article of manufacture comprising:

a computer usable medium having a computer readable program code embodied therein, the computer usable medium having:

computer readable program code for identifying a feature set of the source file;

computer readable program code for assembling the feature set in a buffer; and

computer readable program code for writing the feature set into the target file in the target format.

37. (Previously presented) Computer readable program code for translating a source file in a source format to a target file in a target format, the computer readable program code comprising:

computer readable program code for identifying a feature set of the source file;

computer readable program code for assembling the feature set in a buffer; and

computer readable program code for writing the feature set into the target file in the target format.

38. (Previously presented) A translator, embodied in a computer readable storage medium, for translating a source file in an MIF format to a target file in an HTML format, the translator comprising:

a feature identifier having a front-end lookup table to map MIF code fragments of the source file to a list of features to determine a feature set of the source file;

a buffer to store and assemble the feature set;

a feature writer having a back-end lookup table to map the feature set to HTML code fragments, to write the feature set into the target file in the HTML format; and

an output module to output the target file.